

2

WRDC-TR-89-3040

AD-A208 446

A COMPOUND SCALING ALGORITHM FOR MATHEMATICAL OPTIMIZATION

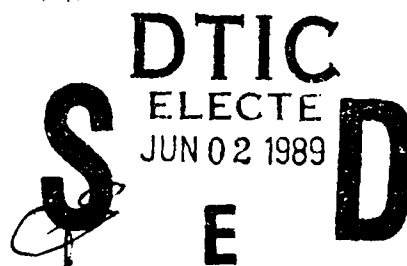


V.B. Venkayya

V.A. Tischler

Analysis and Optimization Branch
Structures Division

February 1989



Final Report for the Period November 1988 to February 1989

Approved for public release; distribution is unlimited.

FLIGHT DYNAMICS LABORATORY
WRIGHT RESEARCH and DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6523

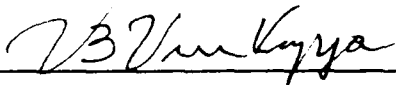
89 6 01 054

NOTICE

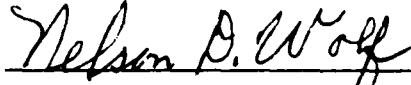
When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise as in any manner, as licensing the holder or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.



VIPPERLA B. VENKAYYA
Project Engineer
Design & Analysis Methods Group



NELSON D. WOLF, Technical Manager
Design & Analysis Methods Group
Analysis & Optimization Branch

FOR THE COMMANDER



JOHN T. ACH, Chief
Analysis Optimization Branch
Structures Division

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/FIBRA, WPAFB, OH 45433-6553 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

| REPORT DOCUMENTATION PAGE | | | | Form Approved OMB No. 0704-0188 | |
|---|-------|---|---|--|--|
| 1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED | | | 1b. RESTRICTIVE MARKINGS | | |
| 2a. SECURITY CLASSIFICATION AUTHORITY | | | 3. DISTRIBUTION/AVAILABILITY OF REPORT | | |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE | | | Approved for public release; distribution is unlimited. | | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S) WRDC-TR-89-3040 | | | 5. MONITORING ORGANIZATION REPORT NUMBER(S) | | |
| 6a. NAME OF PERFORMING ORGANIZATION Analysis & Optimization Branch Structures Division | | 6b. OFFICE SYMBOL (if applicable) | 7a. NAME OF MONITORING ORGANIZATION | | |
| 6c. ADDRESS (City, State, and ZIP Code) WRDC/FIBRA Wright-Patterson AFB OH 45433-6553 | | 7b. ADDRESS (City, State, and ZIP Code) | | | |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION Flight Dynamics Laboratory | | 8b. OFFICE SYMBOL (if applicable) WRDC/FIBRA | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N/A | | |
| 8c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB OH 45433-6553 | | 10. SOURCE OF FUNDING NUMBERS | | | |
| | | PROGRAM ELEMENT NO. 61102F | PROJECT NO. 2302 | TASK NO. N5 | WORK UNIT ACCESSION NO. 06 |
| 11. TITLE (Include Security Classification) A Compound Scaling Algorithm for Mathematical Optimization | | | | | |
| 12. PERSONAL AUTHOR(S) Vipperla B. Venkayya, Victoria A. Tischler | | | | | |
| 13a. TYPE OF REPORT Final | | 13b. TIME COVERED FROM Nov 88 To Feb 89 | | 14. DATE OF REPORT (Year, Month, Day) 1989, February | |
| 15. PAGE COUNT 47 | | | | | |
| 16. SUPPLEMENTARY NOTATION | | | | | |
| 17. COSATI CODES | | | 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) | | |
| FIELD | GROUP | SUB-GROUP | | | |
| 12 | 01 | | | | |
| 19. ABSTRACT (Continue on reverse if necessary and identify by block number) This paper derives a compound scaling algorithm from a simple scaling algorithm for use with general mathematical optimization problems. Compound scaling is necessary when the constraints are nonlinear and the variables do not all belong to the active set. Compound scaling in conjunction with the optimality criteria method has practical applications in large scaled multidisciplinary design. It will extend the scope of the optimality criteria method to handle problems with thousands of variables and constraints. The development of simple scaling is included as a basis for the development of the compound scaling algorithm. Five problems were selected to demonstrate the effectiveness of the simple and compound scaling algorithms. <i>(ks) (—)</i> | | | | | |
| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS | | | 21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED | | |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL Vipperla B. Venkayya | | | 22b. TELEPHONE (Include Area Code) 513-255-7191 | | 22c. OFFICE SYMBOL WRDC/FIBR |

FOREWORD

This report is prepared as part of in-house research under basic research Project 2302, Task No. 2302N5, "Structural Dynamics and Controls," and Work Unit 2302N506 of the same title. The work was carried out in the Design and Analysis Methods Group of the Analysis and Optimization Branch (FIBR), Structures Division, Flight Dynamics Laboratory of the Wright Research and Development Center (WRDC) at Wright-Patterson AFB, Ohio 45433-6523.

The authors greatly appreciate and acknowledge the support and encouragement of Mr. Nelson Wolf, Technical Manager of the Design and Analysis Methods Group.

The report was reviewed by a technical committee headed by Maj R. Hinrichsen and supported by two committee members, Maj L. Hudson and Capt R. Canfield. Their thorough review and suggestions significantly improved the clarity of the report. The authors sincerely thank the committee for its effort.

The time period of the research for this Technical Report is November 1988 - February 1989. The manuscript was released for publication in February 1989.

| | |
|--------------------|-------------------------------------|
| Accession For | |
| NTIS GRA&I | <input checked="" type="checkbox"/> |
| DTIC TAB | <input type="checkbox"/> |
| Unannounced | <input type="checkbox"/> |
| Justification | |
| By _____ | |
| Distribution/ | |
| Availability Codes | |
| Dist | Avail and/or Special |
| A-1 | |



TABLE OF CONTENTS

| SECTION | TITLE | PAGE |
|---------|--|------|
| 1.0 | INTRODUCTION | 1 |
| 2.0 | MATHEMATICAL STATEMENT OF THE PROBLEM | 3 |
| 3.0 | DEVELOPMENT OF SIMPLE SCALING | 5 |
| 3.1 | CLASSIFICATION OF CONSTRAINTS | 5 |
| 3.1.1 | POSITIVE CONSTRAINTS $z_j \geq 0$ | 6 |
| 3.1.2 | NEGATIVE CONSTRAINTS $z_j < 0$ | 7 |
| 3.1.3 | CLASSIFICATION BASED ON A MARGIN OF SAFETY (MS) DEFINITION | 8 |
| 3.2 | SCALE FACTOR DERIVATION FOR SIMPLE SCALING | 9 |
| 3.2.1 | CASE 1 | 10 |
| 3.2.2 | CASE 2 | 13 |
| 3.3 | INTERACTION FORMULA FOR THE SCALE FACTOR | 13 |
| 3.3.1 | CASE 1 | 18 |
| 3.3.2 | CASE 2 | 18 |
| 3.3.3 | LIMITATIONS | 18 |
| 3.3.4 | OPTIONS | 19 |
| 4.0 | COMPOUND SCALING ALGORITHM | 22 |
| 4.1 | CONSTRAINT VALUES MATRIX - $\bar{z}_{g \times 1}$ | 22 |
| 4.2 | CONSTRAINT GRADIENT MATRIX - $N_{n \times s}$ | 23 |
| 4.3 | GENERATION OF THE μ PARAMETERS | 23 |
| 4.3.1 | CASE 1 | 24 |
| 4.3.2 | CASE 2 | 24 |
| 4.3.3 | EXCEPTIONS | 24 |
| 4.4 | SCALE FACTOR TABLE | 25 |
| 4.5 | SCALE FACTOR ASSIGNMENT TABLE | 26 |
| 4.6 | APPROPRIATE SCALE FACTOR FOR EACH VARIABLE | 27 |
| 5.0 | RESULTS OF APPLICATIONS | 29 |
| 6.0 | SUMMARY AND CONCLUSIONS | 44 |

LIST OF ILLUSTRATIONS

| <u>FIGURE</u> | <u>TITLE</u> | <u>PAGE</u> |
|---------------|--|-------------|
| 1 | Two Variable Space | 4 |
| 2 | Constraint Classification for Positive Constraints | 6 |
| 3 | Constraint Classification for Negative Constraints | 7 |

LIST OF TABLES

| TABLE | TITLE | PAGE |
|-------|--|------|
| 1 | Format of the Constraint Gradient Table | 23 |
| 2 | Format of the Scale Factor Table | 25 |
| 3 | Format of the Scale Factor Assignment Table | 26 |
| 4 | PROBLEM 5: Constraint Gradient Matrix for Violated Constraints | 40 |
| 5 | PROBLEM 5: Scale Factor Table | 41 |
| 6 | PROBLEM 5: Scale Factor Assignment Table | 41 |
| 7 | PROBLEM 5: Values of the Variables Before and After Scaling | 42 |
| 8 | PROBLEM 5: Values of the Constraints Before and After Scaling | 43 |

1.0 INTRODUCTION

Constrained minimization/maximization problems are of interest in many engineering disciplines as well as mathematics and physics. This problem, often described as optimization, consists of a performance (objective) function to be minimized and a number of inequality and equality functions which define the region of interest (constraint boundary of the region). Both the objective and constraints are functions of a common set of variables. Some of the constraints are derived from the system performance bounds, and others are simply the limits on the variables themselves. This optimization problem is of such generic interest that it has attracted significant attention in the research community over the last forty years. As a result, a number of highly sophisticated linear and nonlinear programming algorithms are available at present for a variety of applications. Notwithstanding these developments, there are serious shortcomings in the way of applications to *large scale multidisciplinary design*. The optimality criteria method⁽¹⁻⁴⁾ has the potential to extend optimization to problems with thousands of variables and constraints. One of the significant elements of the optimality criteria method is the concept of scaling. Scaling implies changing the variables with the objective of bringing the constraints to the boundary. The constraints can be brought to their boundaries either by adding differential quantities or by multiplying the variables by scale factors. The latter is called scaling. This procedure was originally proposed for constraints which could be scaled in a single step^(1,2). Later it was generalized for nonlinear functions^(3,4). In both cases, however, a simple scaling algorithm was used, and it was adequate for structural optimization problems. On simple scaling a dominant constraint can be identified, and scaling with this constraint generally brings the remaining constraints into the feasible region. This is not necessarily the case in general mathematical optimization problems, and a compound scaling is necessary in order to extend the scope of the optimality criteria method. Another significant element of the optimality criteria method is modification of the variable vector at the constraint

boundary by directly invoking the optimality as defined by the Kuhn-Tucker conditions⁽⁴⁾.

The object of this paper is to give a comprehensive derivation of the compound scaling algorithm with examples to illustrate its effectiveness in reaching the constraint surface or the intersection of the constraints. The details of its application in optimization are given later⁽⁵⁾. It is worthwhile pointing out, however, that it is tacitly assumed that the optimum in constrained minimization (or maximization) problems lies on the constraint boundary or at the intersection of the constraint boundaries. All other cases belong to unconstrained minimization problems, and they are not relevant to the present paper.

It is often necessary in optimization problems to scale the constraint functions to their boundary, since most optimal solutions in constrained minimization (or maximization) problems are at the boundary of the feasible and nonfeasible regions. This can be accomplished by a numerical search in the context of nonlinear programming. However, this numerical search can be tedious and time consuming. A simple scaling algorithm, without formal mathematical derivation, was first used in 1968⁽¹⁾ to locate the constraint boundary in a single step. This simple scaling algorithm was generalized in 1988^(3,4) to include constraints that require more than one scaling step. A formal mathematical derivation, starting with a first order Taylor Series approximation, was presented^(3,4). The algorithm was still a simple scaling, because it was assumed that scaling with a single dominant constraint function would satisfy the remaining constraints. This is the case in most structural optimization problems when all the variables belong to an active set. References (3) and (4) define active and passive variables as well as active and passive constraints.

2.0 MATHEMATICAL STATEMENT OF THE PROBLEM

The constrained optimization problem can be stated mathematically as follows:

Minimize or maximize a function $F(x)$

$$F(x) = F(x_1, x_2, \dots, x_n) \quad (1)$$

subject to the inequality constraints

$$z_j(x_1, x_2, \dots, x_n) \leq z_j \quad j = 1, 2, \dots, k \quad (2)$$

and equality constraints

$$z_j(x_1, x_2, \dots, x_n) = z_j \quad j = k + 1, \dots, s \quad (3)$$

The basic procedure for the solution of this numerical optimization problem is to devise a search strategy to find the optimum, starting from a given initial solution. Most successful search techniques are generally based on the constraint and objective function gradient information. This gradient information is most meaningful (effective) when it is determined at the constraint boundary (separation between the feasible and the infeasible regions - Fig.1), because the optimum is usually on this boundary in constrained minimization (maximization) problems. As a result, constraint boundary location is extremely important. An efficient optimization algorithm should have the facility to reach this boundary at a minimum computational cost from any point in the n -dimensional space. The purpose of the compound scaling algorithm is to provide such facility.

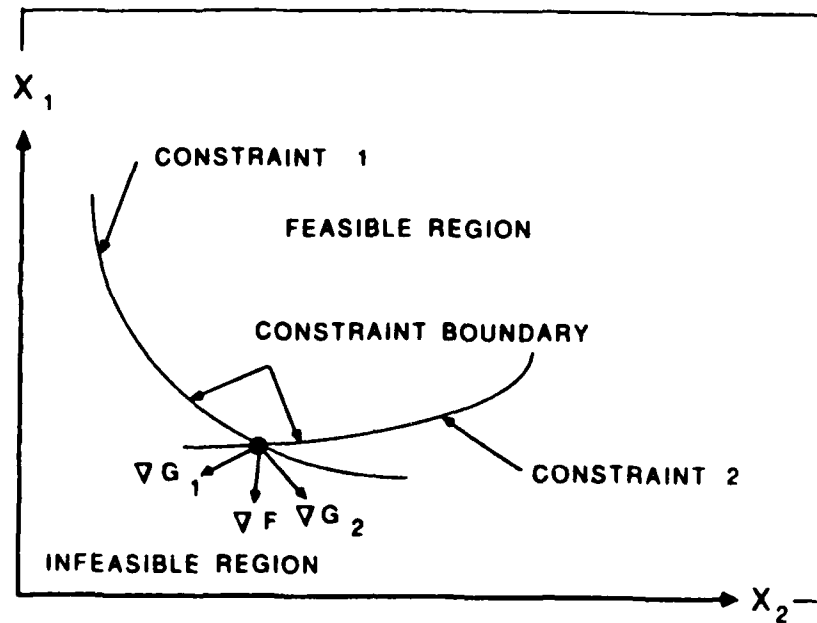


Fig1 : TWO VARIABLE SPACE

The details of how to use this scaling algorithm for the solution of the optimization problem are given in Reference 5. The discussion in this paper is limited to the following problem. Given an initial solution \bar{x} , how to scale \bar{x} such that none of the constraints are violated while at least one of the constraints is at the boundary.

3.0 DEVELOPMENT OF SIMPLE SCALING

The underlying assumption of simple scaling is that there exists a dominant (most violated) constraint and scaling with respect to this constraint will bring all the other constraints to the feasible region. This is not true, in general, when the constraints are nonlinear or when the variables do not all belong to an active set. In such cases a compound scaling algorithm is necessary and it will be discussed in the next section. However, the simple scaling is still the basis for the development of the compound scaling algorithm, and its details are presented here for proper understanding. The simple scaling procedure was discussed earlier^(3,4) but will be repeated here for completeness. Classification of constraints based on their relationship to the constraint boundary precedes the application of the scaling algorithm. The next subsection discusses the procedure for classification.

3.1 Classification of Constraints

The initial variable vector, x , is assumed to be given. Evaluation of all the constraints with this variable vector results in three cases:

- CASE 1: Some constraints are violated.
- CASE 2: None of the constraints are violated.
- CASE 3: Some constraints are at the boundary and others are not violated.

The first two cases are the candidates for scaling, and it is usually not necessary to scale in the third case.

The three cases can be discerned in an easy way by examining a set of β parameters or target response ratios. The β parameter is defined as

$$\beta_j = \frac{z_j}{z_j} \quad (4)$$

where the subscript j indicates the j^{th} constraint (Eqs. 2, 3). \bar{z}_j is the limiting value of the constraint. z_j is the computed value of the constraint with the given or the assumed variable vector, \underline{z} . Based on the β values the constraints can be classified into three categories:

- Violated constraints - Infeasible region
- Active constraints - On the constraint boundary
- Inactive or passive constraints - Feasible region

The β values for these three cases are defined for various situations.

3.1.1 Positive Constraints $\bar{z}_j \geq 0$

For positive constraints four cases can arise:

1. $0 < \beta_j < 1$ Violation for all Positive Constraints
2. $\beta_j = 1$ Active Constraint
3. $\beta_j > 1$ Inactive or Passive Constraint
4. $\beta_j \leq 0$ Inactive or Passive Constraint

These four cases are illustrated in Fig. 2.

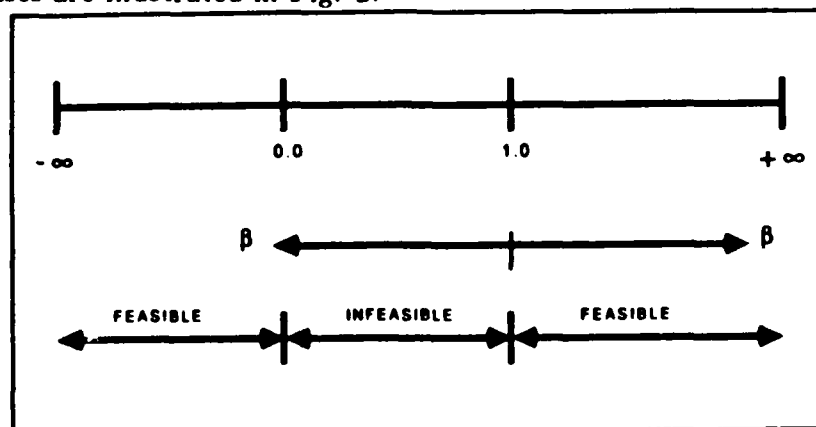


Fig :2 CONSTRAINT CLASSIFICATION
POSITIVE CONSTRAINTS

3.1.2 Negative Constraints $\bar{z}_j < 0$

Here again four cases can arise:

1. $\beta_j > 1$ Violation for all Negative Constraints - Infeasible
2. $\beta_j < 0$ Violation for all Negative Constraints - Infeasible
3. $0 < \beta_j < 1$ Inactive or Passive Constraint - Feasible
4. $\beta_j = 1$ Active Constraint

These four cases are illustrated in Fig. 3.

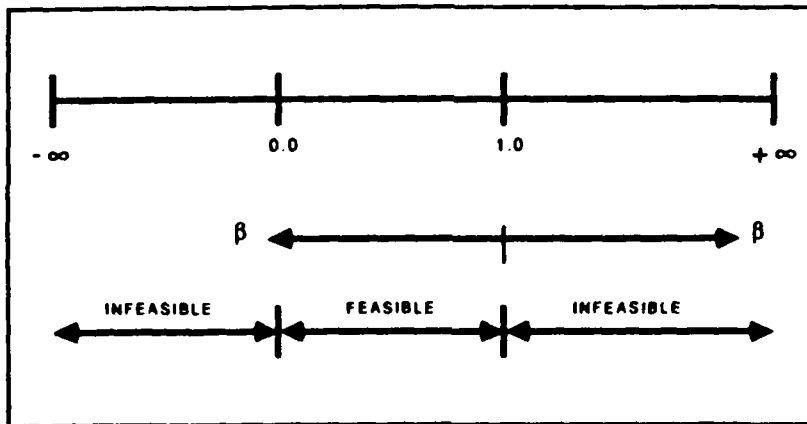


Fig :3 CONSTRAINT CLASSIFICATION
NEGATIVE CONSTRAINTS

The following rules are recommended in order to facilitate the formulation and implementation of the algorithm. The first two rules pertain to the formulation, and the last two rules to the implementation.

Rule 1: The first step in classifying constraints is to formulate all the inequality constraints as

$$z_j \leq \bar{z}_j \quad (5)$$

If there are constraints in the form $z_j \geq z_j$, multiply both sides of the inequality by -1 . Then these inequalities reduce to $-z_j \leq -z_j$.

Rule 2: If z_j is zero, then take one of the terms from the left side of the inequality to the right. If there is only one term on the lefthand side, then let $z_j = \epsilon$ where $|\epsilon| \neq 0$.

Rule 3: If z_j becomes zero, then write the inequality as

$$z_j^{old} - \frac{z_j^{old}}{2} \leq \frac{z_j^{old}}{2} \quad (6)$$

Now the new $z_j^{new} = z_j^{old} - \frac{z_j^{old}}{2}$ and the new $z_j^{new} = \frac{z_j^{old}}{2}$. The new z_j and z_j are used in the β definition.

Rule 4: If β is zero or negative, assume a small value (positive) for β (.1 or less) and scale first. Then evaluate the constraint and continue scaling until β becomes positive.

3.1.3 Classification Based on a Margin of Safety (MS) Definition

It is sometimes more convenient to classify the constraints based on the definition of margin of safety (MS). The original inequality constraint is given as

$$z_j \leq z_j \quad (7)$$

This inequality can be redefined in terms of MS as

$$M.S. = \left(\frac{z_j}{z_j} - 1 \right) \geq 0 \quad (8)$$

Now if z_j/z_j is defined as β , then

$$M.S. = (\beta_j - 1) \geq 0 \quad (9)$$

Now three possible cases arise:

CASE 1: $\beta > 1$ Positive MS - Feasible Region

CASE 2: $\beta = 1$ At the Constraint Boundary

CASE 3: $\beta < 1$ Negative MS - Infeasible Region

If all the constraints have a positive MS, then it is advantageous to bring the nearest constraints to their boundary by scaling. If some of the constraints have a negative MS, then it is necessary to bring them to the constraint boundary by scaling. When some of the constraints are at their boundary and others are all with a positive MS, then no scaling is necessary.

3.2 Scale Factor Derivation For Simple Scaling

The first objective of scaling is to bring all the violated constraints to the feasible region or to the constraint boundary. If all the constraints are in the feasible region, then scaling can be used to bring some of the constraints to the constraint boundary. The mathematical basis for scaling can be derived from the first order approximation of a Taylor's series.

The vector, \underline{x} , is the given or assumed vector of variables. $\underline{\tilde{x}}$ is the scaled vector to bring the violated constraints to the feasible region. The relationship between these two vectors in simple scaling is assumed to be

$$\underline{\tilde{x}} = \Lambda \underline{x} \quad (10)$$

where Λ is a scalar multiplier for all the variables. Let the differential vector $d\underline{x}$ be defined as

$$d\underline{x} = \underline{\tilde{x}} - \underline{x} = (\Lambda - 1)\underline{x} \quad (11)$$

If z_j is the constraint function, the first order approximation of a change in z_j due to a change in x can be written as

$$dz_j = \frac{\partial z_j}{\partial x_1} dx_1 + \frac{\partial z_j}{\partial x_2} dx_2 + \cdots + \frac{\partial z_j}{\partial x_n} dx_n \quad (12)$$

Substitution of (11) into (12) gives an expression for dz_j as

$$dz_j = (\Lambda - 1) \sum_{i=1}^n N_{ij} x_i \quad (13)$$

where n in the summation represents the number of variables participating in the change. It is assumed in this derivation that all the variables are participating. The constraint gradient N_{ij} (differentiation of the j^{th} constraint with respect to the i^{th} variable) is defined as

$$N_{ij} = \frac{\partial z_j}{\partial x_i} \quad (14)$$

Now Eq. 13 can be written as

$$\frac{dz_j}{z_j} = (\Lambda - 1) \sum_{i=1}^n \frac{N_{ij} x_i}{z_j} \quad (15)$$

An examination of Eq. 15 presents two interesting cases which will be discussed in detail in the next two subsections.

3.2.1 Case 1

The first case, henceforth, will be referred to as the active case for distinction. This designation comes from the discussion of active and passive variables in References 3 and 4. This case corresponds to

$$\sum_{i=1}^n \frac{N_{ij} x_i}{z_j} < 0 \quad (16)$$

In this case a parameter μ_{jA} is defined as

$$\mu_{jA} = - \sum_{i=1}^n \frac{N_{ij} x_i}{z_j} \quad (17)$$

With this definition Eq. 15 can be written as

$$\frac{dz_j}{z_j} = (1 - \Lambda_{jA})\mu_{jA} \quad (18)$$

Now the scale factor Λ_{jA} can be written as

$$\Lambda_{jA} = 1 - \frac{dz_j}{z_j} \frac{1}{\mu_{jA}} = 1 - b_j \quad (19)$$

where b_j is defined as

$$b_j = \frac{1}{\mu_{jA}} \frac{dz_j}{z_j} \quad (20)$$

Eq. 19 can also be written as

$$\Lambda_{jA} = \frac{1}{1 - b_j} \simeq 1 + b_j \quad (21)$$

by neglecting the higher order terms of b in a binomial expansion, provided

$$|b_j| \ll 1 \quad (22)$$

From Eq. 21 $\frac{dz_j}{z_j}$ can be written as

$$\frac{dz_j}{z_j} = \frac{\mu_{jA}}{\Lambda_{jA}} - \mu_{jA} \quad (23)$$

or

$$\frac{z_j + dz_j}{z_j} = \frac{\mu_{jA}}{\Lambda_{jA}} - \mu_{jA} + 1 \quad (24)$$

The constraint value z_j corresponds to the variable vector, \underline{x} , and $z_j + dz_j$ corresponds to the new variable vector \underline{x} . If the object is to bring the constraint to the boundary, then it can be written

$$z_j \simeq z_j + dz_j \quad (25)$$

From the definition of the β parameter (in Eq. 4), Eq. 24 can be written as

$$\beta_j = \frac{\mu_{jA}}{\Lambda_{jA}} - \mu_{jA} + 1 \quad (26)$$

Solving for the scale factor, Λ_{JA} , from Eq. 26 gives

$$\Lambda_{JA} = \frac{\mu_{JA}}{\beta_J + \mu_{JA} - 1} \quad (27)$$

Now \tilde{x} can be written as

$$\tilde{x} = \Lambda_{JA} x \quad (28)$$

The approximations indicated in Eqs. 21 and 25 are contingent upon two conditions. The first condition is the first order approximation of a Taylor's series which limits the application to small changes. The second condition is contingent upon the parameter $|b|$ being very small compared to 1. From Eqs. 26 and 21, b can be written as

$$b = \frac{1}{\mu_{JA}} (\beta_J - 1) \quad (29)$$

OBSERVATION 1: It is interesting to note that for the case of $\mu_{JA} = 1$, the scale factor, Λ_{JA} , given by

$$\Lambda_{JA} = \frac{1}{\beta_J} \quad (30)$$

is exact regardless of the range of β_J . The first order approximation in the Taylor's series expansion and the inverse first order approximation in the binomial expansion, Eq. 21, appear to eliminate the approximation altogether at least for the case of $\mu_{JA} = 1$.

OBSERVATION 2: If all n terms in the summation in Eq. 17 are of the same sign and the degree of nonlinearity is > 1 , then

$$\mu_{JA} \geq 1 \quad (31)$$

OBSERVATION 3: The parameter μ is a measure of nonlinearity of the constraint functions. This fact was used to construct an interaction formula⁽⁴⁾ which will be generalized later in this section.

3.2.2 Case 2

The second case, designated as the passive case, is when

$$\sum_{i=1}^n \frac{N_{ij}x_i}{z_j} \geq 0 \quad (32)$$

A μ parameter is then defined as

$$\mu_{jP} = \sum_{i=1}^n \frac{N_{ij}x_i}{z_j} \quad (33)$$

Then the scale factor Λ_{jP} can be written as

$$\Lambda_{jP} = \frac{\beta_j + \mu_{jP} - 1}{\mu_{jP}} \quad (34)$$

OBSERVATION 4: The scale factor as derived from Eq. 34 is only valid for small changes, consistent with the first order Taylor series approximation, except when $\mu_{jP} = 1$.

OBSERVATION 5: The scale factor in Case 1 is inversely proportional, while the scale factor in Case 2 is directly proportional to the target response ratio, β .

The second subscripts A and P in the definitions of the parameters μ and Λ are simply for the convenience of distinguishing Cases 1 and 2. This distinction is very crucial for identifying active and passive variables in scaling. The main point is not which case represents the active or passive condition, but instead it is to recognize that these two are distinctly different cases, and Case 1 should be handled as an inversely proportional condition while Case 2 is a directly proportional condition.

3.3 Interaction Formula for the Scale Factor

The scale factors derived from the first order approximation of a Taylor's series, Eqs. 27 and 34, represent an interesting generalization for nonlinear functions. However, the nature of approximation is such that its usefulness is limited to a very narrow range of

about 10 to 15% on either side of the series expansion (β value of 0.85 to 1.15). Beyond this range the error of approximation is not acceptable, because it may take too many scaling steps to reach the constraint boundary. To overcome this difficulty an interesting interaction formula was proposed⁽⁴⁾ in 1988. The purpose of this interaction formula is to extend the range of application of the scaling algorithm from $-\infty$ to ∞ , while retaining most of the accuracy accustomed to in linear scaling. After a brief discussion of the motivation behind this interaction formula, a formal generalization will be presented with the help of simple mathematical functions.

A formula involving linear and nonlinear interaction is written⁽⁴⁾ for the special case of a beam where axial deformation represents a linear condition and bending deformation a nonlinear condition. An interaction formula is based on the assumption that exact scaling formulas are available at the two extreme conditions: i.e. when the beam deformation is entirely due to axial forces, the scaling formula is given by

$$\Lambda \propto \frac{1}{\beta} \quad (35a)$$

The other extreme is when the beam deformation is entirely due to bending. Then the scaling formula is given by

$$\Lambda \propto \left(\frac{1}{\beta} \right)^n \quad (35b)$$

where the parameter n is a measure of nonlinearity represented by the bending. However, when the actual deformation is a result of both the axial and bending forces, then an approximate interaction formula can be written as

$$\Lambda = \frac{\mu_{A_j}}{\mu_{A_j}} \left(\frac{1}{\beta} \right) + \frac{\mu_{B_j}}{\mu_{B_j}} \left(\frac{1}{\beta} \right)^n \quad (35c)$$

For the notation of the parameters in Eq. 35c see Reference 4.

The basis for generalization of the scaling algorithm can be established by examining

the μ and β parameters of two simple constraint functions

$$z_1 = \frac{c_1}{x_1} + \frac{c_2}{x_2} + \cdots + \frac{c_n}{x_n} \leq z_1 \quad (36)$$

$$z_2 = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \leq z_2 \quad (37)$$

where x_1, x_2, \dots, x_n are the variables and c_1, c_2, \dots, c_n are a set of positive constants. The quantities z_1 and z_2 are the limits of the inequalities. The μ parameters in these two cases are defined as

$$\mu_{1A} = - \sum_{i=1}^n \frac{N_{i1} x_i}{z_1} = 1 \quad (38)$$

$$\mu_{2P} = \sum_{i=1}^n \frac{N_{i2} x_i}{z_2} = 1 \quad (39)$$

Interestingly, the measure of nonlinearity in both these cases is 1, and it represents the linear case. Now the parameters β are defined as

$$\beta_1 = \frac{z_1}{z_1} \quad (40)$$

$$\beta_2 = \frac{z_2}{z_2} \quad (41)$$

It is easy to show that the exact scale factors in these two cases are simply

$$\Lambda_{1A} = \frac{1}{\beta_1} \quad (42)$$

$$\Lambda_{2P} = \beta_2 \quad (43)$$

The first condition represents Case 1 and the second represents Case 2 in the earlier discussion in Sections 3.2.1 and 3.2.2, respectively. A similar examination of two nonlinear functions

$$z_1 = \frac{c_1}{x_1^m} + \frac{c_2}{x_2^m} + \cdots + \frac{c_n}{x_n^m} \quad (44)$$

$$z_2 = c_1 x_1^m + c_2 x_2^m + \cdots + c_n x_n^m \quad (45)$$

gives an interesting result. The c 's are again assumed to be positive constants. The μ parameters for these two cases are

$$\mu_{1A} = m \quad (46)$$

$$\mu_{2P} = m \quad (47)$$

The measure of nonlinearity in these two cases is m . It is easy to again show that the exact scale factors for these two cases are

$$\Lambda_{1A} = \left(\frac{1}{\beta_1} \right)^{\frac{1}{\mu_{1A}}} \quad (48)$$

$$\Lambda_{2P} = (\beta_2)^{\frac{1}{\mu_{2P}}} \quad (49)$$

Now an examination of constraint functions with both linear and nonlinear terms

$$z_1 = \frac{c_1}{x_1} + \frac{c_2}{x_2} + \cdots + \frac{c_n}{x_n} + \frac{c_{n+1}}{x_1^m} + \cdots + \frac{c_{2n}}{x_n^m} \leq 1 \quad (50)$$

$$z_2 = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n + c_{n+1} x_1^m + \cdots + c_{2n} x_n^m \leq 1 \quad (51)$$

gives the basis for the interaction formula derived intuitively earlier⁽⁴⁾ (Eq. 35c). The parameter μ in these cases is no longer a constant, but instead it straddles between limits 1 and m , i.e.,

$$1 \leq \mu \leq m \quad (52)$$

This means that the measure of nonlinearity varies between the two extremes depending on the values of the variables. For example, when the values of the x 's are very large

$$x_i \gg 1 \quad i = 1, 2, \dots, n \quad (53)$$

the value of the μ parameter for the the first constraint approaches the lower limit 1. Conversely, it approaches the upper limit m when the x_i 's are very small.

$$x_i \ll 1 \quad i = 1, 2, \dots, n \quad (54)$$

The second constraint, Eq. 51, represents the opposite case in the sense that the μ parameter would be 1 for small values of x_i and m for large values. It is assumed that the elements of the normalized ξ vector are not too far from one.

The suggested interaction formulae in these two cases are

$$\Lambda_{1A} = a_1 \left(\frac{1}{\beta_1} \right) + a_2 \left(\frac{1}{\beta_1} \right)^{\frac{1}{m}} \quad (55)$$

$$\Lambda_{2P} = a_1 (\beta_2) + a_2 (\beta_2)^{\frac{1}{m}} \quad (56)$$

where the first term represents the linear contribution while the second is the nonlinear part. However, it turns out that these interaction formulae are equivalent to

$$\Lambda_{1A} = \left(\frac{1}{\beta_1} \right)^{\mu_{1A}} \quad (57)$$

$$\Lambda_{2P} = (\beta_2)^{\mu_{2P}} \quad (58)$$

The scale factors given by Eqs. 57 and 58 are far superior approximations than those given by Eqs. 27 and 34. The range of their application is not limited to 10 to 15% on either side of $\beta = 1$. However, they are not exact when the summation in the μ parameter definition contains both positive and negative terms. In such cases two scale factors are defined for each constraint. These correspond to Case 1 and Case 2 discussed earlier. For example, the two scale factors for the j^{th} constraint can be defined as

3.3.1 Case 1

$$\Lambda_{JA} = \left(\frac{1}{\beta_J} \right)^{\mu_{JA}} \quad (59)$$

3.3.2 Case 2

$$\Lambda_{JP} = \left(\beta_J \right)^{\mu_{JP}} \quad (60)$$

The definition of μ_{JA} and μ_{JP} in Eq. 59 and 60 is as follows:

$$\mu_{JA} = - \sum_{i=1}^n \frac{N_{ij} x_i}{z_j} \quad (17a)$$

and

$$\mu_{JP} = \sum_{i=1}^n \frac{N_{ij} x_i}{z_j} \quad (33a)$$

The important distinction between the original equations (17 and 33) and their alternates (17a and 33a) is that the two summations include only terms of the same sign. In other words the sum in Eq. 17a includes only negative terms and similarly Eq. 33a contains only positive terms.

The scale factor Λ_{JA} is applicable to all the variables that contribute to the sum in Eq. 17a. Similarly, the scale factor Λ_{JP} is applicable to the variables in the sum in Eq. 33a.

3.3.3 Limitations

The following limitations apply to the scale factor calculated from Eqs. 59 and 60.

1. If $\Lambda_{JA} \gg 1$ or $\ll 1$, then no scaling should be allowed, because such a condition represents a pathological or asymptotic condition, and the scaling would be

counter-productive. In such cases the default is $\Lambda_{JA} = 1$ which represents no scaling.

2. The same rule applies to Λ_{JP} as well.
3. A reasonable quantitative limit for Λ_J is as follows (in order to avoid large changes in the variables):

$$0.01 < \Lambda_{JA}(\text{or } \Lambda_{JP}) < 100 \quad (61)$$

These limits are good for the first scaling where the initial variable vector is selected arbitrarily. In subsequent scalings this range can be reduced even further

$$0.1 < \Lambda_{JA}(\text{or } \Lambda_{JP}) < 10 \quad (62)$$

As stated earlier scaling is not allowed outside this range which means Λ_{JA} or $\Lambda_{JP} = 1.0$

This generalization of assigning more than one scale factor leads to the concept of compound scaling where each variable can have a separate scale factor.

3.3.4 Options

When the two scale factors Λ_{JA} and Λ_{JP} are computed for each constraint, two cases can arise:

1. One of the scale factors is out of bounds (as defined by Eqs. 61 and 62), in which case the other scale factor is the only one valid for scaling. Although the possibility of both scale factors being out of bounds is very remote, it indicates that scaling is not possible from that point.
2. Both scale factors are within the bounds in which case they are both valid for scaling. In this case two options must be examined.

OPTION 1: Both scale factors are used for scaling. The active scale factor Λ_{jA} is used for scaling the variables that contributed to the negative sum in the μ parameter definition, while Λ_{jP} is used for the other variables. In such a case, movement towards the feasible or infeasible region is fast and may result in overshooting the constraint boundary. Additional scalings using the two scale factors may result in oscillation about the constraint boundary and in too many scaling steps.

OPTION 2: Only one scale factor is used; *i.e.* only the active or passive variables (corresponding to Λ_{jA} or Λ_{jP}) are allowed to participate in the scaling. The overshooting problem is mitigated in this case. Now the question is which one is a better choice? This decision is based on determining whether this constraint is basically governed by Case 1 or 2 (inversely or directly proportional to β). This determination is quite simple when Λ_{jA} and Λ_{jP} are determined with equal values for all the variables. The largest of the two Λ 's (Λ_{jA} or Λ_{jP}) is the governing condition. This condition (active or passive) remains the same regardless of what happens in subsequent solutions. For example, if both Λ 's are within the bounds in the solution with equal values for all the variables, and Λ_{jA} is the largest of the two, then Λ_{jA} is the governing condition in this and all the subsequent solutions. Scaling for this constraint remains an active case even if Λ_{jP} is greater in subsequent solutions. This conclusion is based on the assumption that when all the variables are equal (with no bias), the μ parameters describe the function behavior more realistically. If the solution is not with equal values for all the variables, then the decision has to be deferred until one of the two scale factors goes out of bounds. Once Λ_{jA} or Λ_{jP} goes out of bounds, it is an indication that it will not be the governing condition for this constraint. If both scale factors remain within the governing condition is by computing Λ_{jA} and Λ_{jP} with equal values for all the variables.

Another important case that needs attention is when β is very small (says $\beta < 0.01$)

and one of the scale factors is out of bounds. In this case the same scale factor should be applied to all the variables. This condition differs from the case when β is not too small ($\beta > 0.01$) and one of the scale factors is out of bounds. In such a case scaling is applied only to the variables belonging to the scale factor in bounds, and the variables belonging to the out of bounds scale factor are left at their current values (no scaling).

So far the discussion has been in the context of a single constraint. An extension of these ideas to the environment of multiple constraints leads to a formal derivation of a compound scaling algorithm in the next section.

4.0 COMPOUND SCALING ALGORITHM

The compound scaling algorithm is a further generalization of the scaling algorithm presented in the previous section. It is assumed that there is no single dominant constraint, and the scaling has to be applied for multiple constraints in order to bring them to the feasible region or to the constraint surface. To put it succinctly, "scaling is a process by which the variables are adjusted such that the constraints are brought to the constraint surface or into the feasible region".

The compound scaling algorithm requires three important tables or rectangular arrays. All three tables (arrays) are of the same dimension n (rows) \times s (columns), where n represents the number of variables and s represents the number of active constraints.

The first table is the constraint gradient matrix, $N_{n \times s}$, which is either given or a facility exists for its generation. It is also assumed that all the constraints can be evaluated for a given variable vector x . To reiterate, the following information is necessary in order to develop the compound scaling algorithm.

1. Constraint values - matrix $z_{s \times 1}$ for a given x
2. Constraint gradient matrix $N_{n \times s}$ for a given x
3. Scale factor table (matrix)
4. Scale factor assignment table (matrix)

The last two tables are generated from the information given in the first two tables. The first two tables are generated from an analysis of the system.

4.1 Constraint Values Matrix - $z_{s \times 1}$

For a given variable vector, x , all the constraints can be evaluated by an analysis of the system. All the constraints that are candidates for scaling are identified. This list includes

all the violated constraints and those close to the constraint surface (within 5 to 10 %). The number of constraints used for scaling is s .

4.2 Constraint Gradient Matrix - $N_{n \times s}$

The elements of the N matrix, N_{ij} , are defined as

$$N_{ij} = \frac{\partial z_j}{\partial x_i} \quad (63)$$

This information is also assumed to be available from an analysis of the system. The format of the constraint gradient matrix is shown below.

TABLE 1 Format of the Constraint Gradient Table

| | | | | | |
|-------------|---|----------|----------|-------|----------|
| Constraints | → | z_1 | z_2 | . . . | z_s |
| Variables | | | | | |
| ↓ | | | | | |
| x_1 | | N_{11} | N_{12} | . . . | N_{1s} |
| x_2 | | N_{21} | N_{22} | . . . | N_{2s} |
| . | | . | . | | . |
| . | | . | . | | . |
| . | | . | . | | . |
| . | | . | . | | . |
| x_n | | N_{n1} | N_{n2} | . . . | N_{ns} |

4.3 Generation of the μ Parameters

From the constraint gradient matrix a set of μ parameters is generated. The μ parameter set consists of $2 \times s$ entries. That is, each constraint is associated with two μ

parameters, and they are defined as follows:

$$\mu_{jA} = - \sum_{i=1}^n \frac{N_{ij} x_i}{z_j} \quad (64)$$

where the above sum includes only negative terms.

$$\mu_{jP} = \sum_{i=1}^n \frac{N_{ij} x_i}{z_j} \quad (65)$$

where the above sum includes only positive terms. The subscript j represents the j^{th} constraint. The second subscripts, A and P , represent the active and passive designation corresponding to Cases 1 and 2, respectively, in simple scaling (reference Eqs. 17 and 33).

Now for each constraint compute two scale factors corresponding to the two μ parameters.

4.3.1 Case 1

$$\Lambda_{jA} = \left(\frac{1}{\beta_j} \right)^{\mu_{jA}} \quad (66)$$

4.3.2 Case 2

$$\Lambda_{jP} = \left(\beta_j \right)^{\mu_{jP}} \quad (67)$$

The interaction formulas discussed in 1988⁽³⁾ are the basis for Eqs. 64 and 65.

4.3.3 Exceptions

- If $\beta > 1$ and $\mu_{jP} \ll 1$ or calculated $\Lambda_{jP} \gg 1$, set $\Lambda_{jP} = 1.0$
- If $\beta < 1$ and $\mu_{jA} \ll 1$ or calculated $\Lambda_{jA} \gg 1$, set $\Lambda_{jA} = 1.0$

- For all other cases use the calculated quantities.

If the scale factor becomes too large, say $\Lambda_j > 100$, it indicates a pathological condition and by setting Λ_{jP} or $\Lambda_{jA} = 1$ the scaling is by-passed for that condition. The scaling is by-passed for that condition. The other conditions can scale the variables.

4.4 Scale Factor Table

The format of the scale factor table is similar to the constraint gradient table.

TABLE 2 Format of the Scale Factor Table

| Constraints \rightarrow | z_1 | z_2 | \dots | z_s |
|---------------------------|---|---|---------|---|
| Variables | | | | |
| \downarrow | | | | |
| x_1 | $\Lambda_{1A} \text{ or } \Lambda_{1P}$ | $\Lambda_{2A} \text{ or } \Lambda_{2P}$ | \dots | $\Lambda_{sA} \text{ or } \Lambda_{sP}$ |
| x_2 | $\Lambda_{1A} \text{ or } \Lambda_{1P}$ | $\Lambda_{2A} \text{ or } \Lambda_{2P}$ | \dots | $\Lambda_{sA} \text{ or } \Lambda_{sP}$ |
| \cdot | \cdot | \cdot | \cdot | \cdot |
| \cdot | \cdot | \cdot | \cdot | \cdot |
| \cdot | \cdot | \cdot | \cdot | \cdot |
| \cdot | \cdot | \cdot | \cdot | \cdot |
| x_n | $\Lambda_{1A} \text{ or } \Lambda_{1P}$ | $\Lambda_{2A} \text{ or } \Lambda_{2P}$ | \dots | $\Lambda_{sA} \text{ or } \Lambda_{sP}$ |

Each of the entries in the j^{th} column is either Λ_{jA} or Λ_{jP} depending on the variable's contribution either to the sum μ_{jA} or μ_{jP} . This table gives s possible scale factors for each variable. Which one of the s scale factors is relevant to the given variable is determined with the help of a scale factor assignment table and the constraint gradient table.

4.5 Scale Factor Assignment Table

The format of the scale factor assignment table is similar to the constraint gradient and scale factor tables.

TABLE 3 Format of the Scale Factor Assignment Table

| Constraints | z_1 | z_2 | \dots | z_s |
|-------------|----------|----------|----------|----------|
| Variables | | | | |
| x_1 | t_{11} | t_{12} | \dots | t_{1s} |
| x_2 | t_{21} | t_{22} | \dots | t_{2s} |
| \vdots | \vdots | \vdots | \vdots | \vdots |
| x_n | t_{n1} | t_{n2} | \dots | t_{ns} |

The entries t_{ij} in this table are calculated one column at a time where

$$t_{ij} = \left| \frac{N_{ij} x_i}{z_j} \right| \quad (68)$$

The parameter μ_{ij} belongs to μ_{jA} or μ_{jP} depending on whether the argument in the absolute value is negative or positive. The table is completed when all the s columns are filled. The values of the entries in the scale factor assignment table vary from zero to one.

4.6 Appropriate Scale Factor for Each Variable

Now all the information is available for the selection of an appropriate scale factor for each variable from the list of s scale factors.

Example: Appropriate scale factor for the i^{th} variable.

Rule 1: From the *scale factor assignment table* select the largest entry in the i^{th} row.

For Example: If t_{ij} is the largest entry in the scale factor assignment table, then the appropriate scale factor for the i^{th} variable is the Λ_{ij} scale factor from the scale factor table.

Rule 2: If there is more than one entry equal to the largest entry, for example: t_{ij} is the largest entry in the scale factor assignment table, but t_{ik} and t_{il} are also equal to t_{ij} . Then the appropriate scale factor is any of the three (Λ_{ij} or Λ_{ik} or Λ_{il}).

Now complete this process of selecting appropriate scale factors for all the variables. Then the scaled variables for x_i are computed by

$$x_i^{new} = x_i^{old} \Lambda_{ij} \quad i = 1, 2, \dots, n \quad (69)$$

where Λ_{ij} is the appropriate scale factor for the i^{th} variable as determined by the two rules.

The next step is to evaluate all the constraints with the new variable vector. If there are any violated constraints, the scaling algorithm is repeated.

The results of the compound scaling algorithm are presented in the section on results when applied to specific examples. These examples do not address the complete optimization problem. Only the scaling aspect is addressed. The complete implementation of the optimization algorithm is given in Reference 5.

5.0 RESULTS OF APPLICATIONS

Five problems were selected to demonstrate the effectiveness of the simple and compound scaling algorithms. The first three problems are well suited for demonstration of the simple scaling algorithm. The last two problems demonstrate application of the compound scaling algorithm.

PROBLEM 1. The inequality constraint function is defined as

$$z_1(x) = \frac{10}{x_1} + \frac{15}{x_2} + \frac{25}{x_3} + \frac{40}{x_4} \leq 1.0$$

The gradients of the constraint function are given by

$$N_{11} = -\frac{10}{x_1^2} \quad N_{21} = -\frac{15}{x_2^2} \quad N_{31} = -\frac{25}{x_3^2} \quad N_{41} = -\frac{40}{x_4^2}$$

CASE 1: The initial solution is assumed to be

$$x_1 = x_2 = x_3 = x_4 = 1.0$$

For this solution $z_1(x) = 90$ which is a significant violation of the constraint. For scaling, the μ parameter is calculated from Eq. 64 as

$$\mu = -\sum_{i=1}^4 \frac{N_{i1} x_i}{z_1} = 1.0$$

which implies that the measure of nonlinearity is 1, as noted earlier. The β parameter is computed by Eq. 4 as

$$\beta = \frac{1}{90} = 0.0111111$$

The scale factor is given by Eq. 66

$$\Lambda = (90)^{1/1} = 90$$

The new variables are given by

$$x_1 = x_2 = x_3 = x_4 = 90$$

For this solution

$$z_1(x) = 1.0$$

The constraint boundary has been reached in a single step, and the scaling is considered to be exact.

CASE 2: The initial solution is assumed to be

$$x_1 = 1 \quad x_2 = 2 \quad x_3 = 3 \quad x_4 = 4$$

For this solution $z_1(x) = 35.8333$ which is a violation of the constraint.

$$\mu = - \sum_{i=1}^4 \frac{N_{i1} x_i}{z_1} = 1.0$$

and

$$\beta = \frac{1}{35.8333} = 0.0279$$

Thus, the scale factor is given by

$$\Lambda = \left(\frac{1}{.0279} \right)^{1/4} \approx 35.8333$$

The new variables are

$$x_1 = 35.8333 \quad x_2 = 71.6667 \quad x_3 = 107.5 \quad x_4 = 143.333$$

For this solution

$$z_1(x) = 1.0$$

Once again the constraint surface has been reached in a single step.

PROBLEM 2. The purpose of this problem is to show that the scaling can be exact under certain conditions, even when the measure of nonlinearity is greater than 1.

The inequality constraint function is defined as

$$z_1(x) = \frac{30}{x_1^3} + \frac{2}{x_2^3} + \frac{108}{x_3^3} + \frac{47}{x_4^3} \leq 1.0$$

The gradients of the constraint function are given by

$$N_{11} = -\frac{90}{x_1^4} \quad N_{21} = -\frac{6}{x_2^4} \quad N_{31} = -\frac{324}{x_3^4} \quad N_{41} = -\frac{141}{x_4^4}$$

The initial solution is assumed to be

$$x_1 = x_2 = x_3 = x_4 = 1.0$$

For this solution $z_1(x) = 187$ which is a violation of the constraint.

$$\mu = - \sum_{i=1}^4 \frac{N_{i1} x_i}{z_1} = 3.0$$

i.e., the measure of nonlinearity is 3.0.

$$\beta = \frac{1}{187} = 0.0053476$$

The scale factor is given by Eq. 66

$$\Lambda = (187)^{1.3} = 5.718479$$

The new variables are

$$x_1 = x_2 = x_3 = x_4 = 5.718479$$

For this solution $z_1(x) = 1.0$

Once again the constraint surface is reached in one step.

PROBLEM 3. The third problem is designed to show that the scaling is not exact when there is a possibility of a variation in the degree of nonlinearity. In such cases scaling

can take more than one step. Still, it would be significantly faster than any search. The constraint function is defined as

$$z_1(x) = \frac{10}{x_1} + \frac{30}{x_1^3} + \frac{15}{x_2} + \frac{2}{x_2^3} + \frac{25}{x_3} + \frac{108}{x_3^3} + \frac{40}{x_4} + \frac{47}{x_4^3} \leq 1.0$$

Gradients of the constraint function are given by

$$N_{11} = -\frac{10}{x_1^2} - \frac{90}{x_1^4}$$

$$N_{21} = -\frac{15}{x_2^2} - \frac{6}{x_2^4}$$

$$N_{31} = -\frac{25}{x_3^2} - \frac{324}{x_3^4}$$

$$N_{41} = -\frac{40}{x_4^2} - \frac{141}{x_4^4}$$

The initial solution is assumed to be

$$x_1 = x_2 = x_3 = x_4 = 1.0$$

For this solution $z_1(x) = 277$ which is a violation of the constraint.

The parameter μ is given by

$$\mu = -\sum_{i=1}^4 \frac{N_{i1}x_i}{z_1} = 2.35018$$

The parameter β is given by

$$\beta = \frac{1}{277} = 0.00361$$

The scale factor from Eq. 66 is

$$\Lambda = (277)^\mu = 10.94645$$

The new variables are given by

$$x_1 = x_2 = x_3 = x_4 = 10.94645$$

For this solution the constraint function $z_1(x) = 8.36441$. The constraint is still violated.

Second Scaling

The constraint gradients with the new variables are given by

$$N_{11} = -0.0897$$

$$N_{21} = -0.1256$$

$$N_{31} = -0.2312$$

$$N_{41} = -0.3436$$

The parameter μ is given by

$$\mu = - \sum_{i=1}^4 \frac{N_{i1} x_i}{z_1} = 1.03409$$

The parameter β is given by

$$\beta = \frac{1}{8.36441} = 0.11955$$

The scale factor is given by

$$\Lambda = (8.36441)^{\frac{1}{\mu}} = 7.79879$$

The new variables are

$$x_1 = x_2 = x_3 = x_4 = 85.3690$$

and the new constraint value for this solution is

$$z_1(x) = 1.05455$$

One more scaling should bring the constraint to the boundary. In the next step μ will be reduced to 1.0. This means that in the first step the cubic terms dominated and gave the μ value as 2.35018, and in the second step the μ value was reduced to 1.03409.

PROBLEM 4. This problem is designed to demonstrate the compound scaling algorithm with a single constraint. The constraint is defined as

$$z_1(x) = 180x_1 + 20x_2 - 3.1x_3 + 0.2x_4 - 5x_1x_2 + 37x_1x_3 + 8.7x_2x_4 - 3x_3x_4 - 0.1x_1^2x_2 + 0.001x_2^2x_3 + 95x_1x_4^2 - 81x_4x_3^2 + x_1^3 - 6.2x_2^3 + 0.48x_3^3 + 22x_4^3 \leq 1.0$$

The constraint gradients are given by

$$\begin{aligned} N_{11} &= 180 - 5x_2 + 37x_3 - 0.2x_1x_2 + 95x_4^2 + 3x_1^2 \\ N_{21} &= 20 - 5x_1 + 8.7x_4 - 0.1x_1^2 + .002x_2x_3 - 18.6x_2^2 \\ N_{31} &= -3.1 + 37x_1 - 3x_4 + 0.001x_2^2 - 162x_4x_3 + 1.44x_3^2 \\ N_{41} &= 0.2 + 8.7x_2 - 3x_3 + 190x_1x_4 - 81x_3^2 + 66x_4^2 \end{aligned}$$

The initial solution is assumed to be

$$x_1 = x_2 = x_3 = x_4 = 1.0$$

For this solution $z_1(x) = 265.981$ which is a violation of the constraint. The constraint gradients are

$$\begin{aligned} N_{11} &= 309.8 \\ N_{21} &= 5.002 \\ N_{31} &= -129.659 \\ N_{41} &= -180.90 \end{aligned}$$

The parameters μ are computed separately for negative, Eq. 64, and positive, Eq. 65, sums.

$$\begin{aligned} \mu_{1A} &= -\frac{N_{31}x_3}{z_1} = 0.487475 \\ \mu_{1P} &= \sum_{i=1,2,4} \frac{N_{i1}x_i}{z_1} = 1.863674 \end{aligned}$$

The parameter β is given by

$$\beta = \frac{z}{z} = 0.003759$$

$$\Lambda_{1A} = \left(\frac{1}{\beta} \right)^{\mu_{1A}} = 94256.43$$

$$\Lambda_{1P} = (\beta)^{\mu_{1P}} = 0.0499905$$

The parameter $\Lambda_{1A} \gg 1$ represents a pathological condition, and it is ignored. So the variable x_3 will not be scaled.

The new variables are

$$x_1 = 0.0499905$$

$$x_2 = 0.0499905$$

$$x_3 = 1.0$$

$$x_4 = 0.0499905$$

The constraint function value is now given as

$$z_1(x) = 5.06174$$

Second Scaling

$$N_{11} = 216.9945$$

$$N_{21} = 20.1383$$

$$N_{31} = -8.0588$$

$$N_{41} = -82.7253$$

$$\mu_{1A} = - \sum_{i=3,4} \frac{N_{i1} x_i}{z_1} = 2.40910$$

$$\mu_{1P} = \sum_{i=1,2} \frac{N_{i1} x_i}{z_1} = 2.34196$$

The parameter β is given by

$$\beta = 0.19756$$

$$\Lambda_{1A} = (5.06175)^{\mu_{1A}} = 1.96042$$

$$\Lambda_{1P} = (0.19756)^{\mu_{1P}} = 0.500344$$

The new variables are

$$x_1 = 0.0250124$$

$$x_2 = 0.0250124$$

$$x_3 = 1.0$$

$$x_4 = 0.0499905$$

The constraint function for these new variables is $z_1(x) = -0.8649050$.

Third Scaling

$$N_{11} = 217.1140992$$

$$N_{21} = 20.29820597$$

$$N_{31} = -8.98296822$$

$$N_{41} = -83.1798822$$

$$\mu_{1A} = - \sum_{i=1,2} \frac{N_{i1} x_i}{z_1} = 6.8657883$$

$$\mu_{1P} = \sum_{i=3,4} \frac{N_{i1} x_i}{z_1} = 15.193772$$

Calculate the β by scaling up to -0.1 in order to avoid a negative β (see Rule 4 sec 3.1.2)

$$\beta = \frac{-0.1}{-0.864905} = 0.1156196$$

$$\Lambda_{1A} = (8.6490503)^{\mu_{1A}} = 1.3692072$$

$$\Lambda_{1P} = (0.1156196)^{\mu_{1P}} = 0.8676250$$

The new variables are:

$$x_1 = 0.02501241$$

$$x_2 = 0.02501241$$

$$x_3 = 0.86762503$$

$$x_4 = 0.04337299$$

The constraint function value is now given as

$$z_1(x) = 0.6929182$$

Fourth Scaling

$$N_{11} = 212.1575314$$

$$N_{21} = 20.2406272$$

$$N_{31} = -7.3169668$$

$$N_{41} = -62.8296117$$

$$\mu_{1A} = - \sum_{i=3,4} \frac{N_{i1} x_i}{z_1} = 13.0946068$$

$$\mu_{1P} = \sum_{i=1,2} \frac{N_{i1} x_i}{z_1} = 8.3889252$$

The parameter β is given by:

$$\beta = 1.4431718$$

$$\Lambda_{1A} = (0.6929182)^{\mu_{1A}} = 0.9723739$$

$$\Lambda_{1P} = (1.4431718)^{\mu_{1P}} = 1.0446997$$

The new variables are:

$$x_1 = 0.0261305$$

$$x_2 = 0.0261305$$

$$x_3 = 0.8676250$$

$$x_4 = 0.0433730$$

The constraint function value is now given as

$$z_1(x) = 0.9527436$$

Fifth Scaling

$$N_{11} = 212.1521012$$

$$N_{21} = 20.2339696$$

$$N_{31} = -7.2755990$$

$$N_{41} = -62.8106710$$

$$\mu_{1A} = - \sum_{i=3,4} \frac{N_{i1}x_i}{z_1} = 9.4850052$$

$$\mu_{1P} = \sum_{i=1,2} \frac{N_{i1}x_i}{z_1} = 6.3735460$$

The parameter β is given by:

$$\beta = 1.0496003$$

$$\Lambda_{1A} = 0.9949092$$

$$\Lambda_{1P} = 1.0076243$$

The new variables are:

$$x_1 = 0.0263297$$

$$x_2 = 0.0263297$$

$$x_3 = 0.8676250$$

$$x_4 = 0.0433730$$

The constraint function value is now

$$z_1(x) = 0.9990408$$

which is within 0.096% of the constraint surface.

PROBLEM 5. This problem, a gear reducer example, is designed to demonstrate the compound scaling algorithm with multiple constraints. A detailed description of the problem is given in Reference 6. In this example there are 7 design variables and 25 constraints.

The constraints are defined as:

$$z_1(x) = \frac{27}{x_1 x_2^2 x_3} \leq 1.0$$

$$z_2(x) = \frac{397.5}{x_1 x_2^2 x_3^2} \leq 1.0$$

$$z_3(x) = \frac{1.93}{x_2 x_3 x_4^3 x_6^4} \leq 1.0$$

$$z_4(x) = \frac{1.93}{x_2 x_3 x_5^3 x_7^4} \leq 1.0$$

$$z_5(x) = \frac{A_1}{B_1} \leq 1100.0$$

$$A_1 = \left(\left(\frac{745 x_4}{x_2 x_3} \right)^2 + (16.9) 10^6 \right)^{1/2}$$

$$B_1 = 0.1 x_6^3$$

$$z_6(x) = \frac{A_2}{B_2} \leq 850.0$$

$$A_2 = \left(\left(\frac{745 x_5}{x_2 x_3} \right)^2 + (157.5) 10^6 \right)^{1/2}$$

$$B_2 = 0.1 x_7^3$$

$$z_7(x) = x_2 x_3 \leq 40.0$$

$$z_8(x) = \frac{x_1}{x_2} \leq -5.0$$

$$z_9(x) = \frac{x_1}{x_2} \leq 12.0$$

$$z_{10}(x) = x_1 \leq 3.6$$

$$z_{11}(x) = -x_1 \leq -2.6$$

$$z_{12}(x) = x_2 \leq 0.8$$

$$z_{13}(x) = -x_2 \leq -0.7$$

$$z_{14}(x) = x_3 \leq 28.0$$

$$z_{15}(x) = -x_3 \leq -17.0$$

$$z_{16}(x) = x_4 \leq 8.3$$

$$z_{17}(x) = -x_4 \leq -7.3$$

$$z_{18}(x) = x_5 \leq 8.3$$

$$z_{19}(x) = -x_5 \leq -7.3$$

$$z_{20}(x) = x_6 \leq 3.9$$

$$z_{21}(x) = -x_6 \leq 2.9$$

$$z_{22}(x) = x_7 \leq 5.5$$

$$z_{23}(x) = -x_7 \leq -5.0$$

$$z_{24}(x) = \frac{(1.5x_6 + 1.9)}{x_4} \leq 1.0$$

$$z_{25}(x) = \frac{(1.1x_7 + 1.9)}{x_5} \leq 1.0$$

The initial solution was selected as the average of the minimum and maximum values defined for each variable: (constraints z_{10} thru z_{23}). For this solution constraints 5, 6, 8, 9 and 10 were violated. Only one scaling was necessary to bring all the violated constraints to the feasible region or to the constraint boundary. The constraint gradient matrix - $N_{7 \times 5}$, the scale factor table and the scale factor assignment table are given for this scaling.

TABLE 4 PROBLEM 5 : Constraint Gradient Matrix for Violated Constraints

| $z \rightarrow$ | 5 | 6 | 8 | 9 | 10 |
|-----------------|-----------|-----------|---------|--------|--------|
| x | | | | | |
| \downarrow | | | | | |
| 1 | 0.0000 | 0.0000 | -1.3333 | 0.0000 | 0.0000 |
| 2 | -9.7511 | -.8703 | 5.5111 | 0.0000 | 0.0000 |
| 3 | -.3250 | -.0290 | 0.0000 | 0.0000 | 0.0000 |
| 4 | .9376 | 0.0000 | 0.0000 | -.1151 | 0.0000 |
| 5 | 0.0000 | .0837 | 0.0000 | 0.0000 | -.1262 |
| 6 | -926.1200 | 0.0000 | 0.0000 | .1923 | 0.0000 |
| 7 | 0.0000 | -495.7786 | 0.0000 | 0.0000 | .1410 |

TABLE 5 PROBLEM 5 : Scale Factor Table

| $z \rightarrow$ | 5 | 6 | 8 | 9 | 10 |
|-----------------|----------|----------|----------|----------|----------|
| r | | | | | |
| ↓ | | | | | |
| 1 | 1.000000 | 1.000000 | 1.209677 | 1.160126 | 1.021703 |
| 2 | .984560 | 1.006856 | .826667 | 1.160126 | 1.021703 |
| 3 | .984560 | 1.006856 | .826667 | 1.160126 | 1.021703 |
| 4 | 1.000000 | 1.000000 | .826667 | .897436 | 1.021703 |
| 5 | 1.000000 | 1.000000 | .826667 | 1.160126 | .983974 |
| 6 | .984560 | 1.000000 | .826667 | 1.160126 | 1.021703 |
| 7 | 1.000000 | 1.006856 | .826667 | 1.160126 | 1.021703 |

TABLE 6 PROBLEM 5 : Scale Factor Assignment Table

| $z \rightarrow$ | 5 | 6 | 8 | 9 | 10 |
|-----------------|----------|----------|----------|----------|----------|
| r | | | | | |
| ↓ | | | | | |
| 1 | 0.000000 | 0.000000 | 1.000000 | 0.000000 | 0.000000 |
| 2 | .006968 | .000752 | 1.000000 | 0.000000 | 0.000000 |
| 3 | .006968 | .000752 | 0.000000 | 0.000000 | 0.000000 |
| 4 | .006968 | 0.000000 | 0.000000 | 1.000000 | 0.000000 |
| 5 | 0.000000 | .000752 | 0.000000 | 0.000000 | 1.000000 |
| 6 | 3.000000 | 0.000000 | 0.000000 | .728571 | 0.000000 |
| 7 | 0.000000 | 3.000000 | 0.000000 | 0.000000 | .752443 |

From the scale factor assignment table, the scale factor table, the constraint gradient table, and the rules enumerated earlier, the appropriate scale factor for each variable was determined. Table 7 gives the values of the appropriate scale factors, the scaled variables and the adjusted variables when they exceeded the limits.

TABLE 7 PROBLEM 5 : Values of the Variables Before and After Scaling

| Variable | Original x 's | Scale Factor | Scaled Variables | Adjusted Variables |
|----------|-----------------|--------------|------------------|--------------------|
| x_1 | .310000E+01 | .120968E+01 | .375000E+01 | .360000E+01 |
| x_2 | .750000E+00 | .826667E+00 | .620000E+00 | .700000E+00 |
| x_3 | .225000E+02 | .984560E+00 | .221526E+02 | .221526E+02 |
| x_4 | .780000E+01 | .897436E+01 | .700000E+01 | .730000E+01 |
| x_5 | .780000E+01 | .983974E+00 | .767500E+01 | .767500E+01 |
| x_6 | .340000E+01 | .984560E+00 | .334750E+01 | .334750E+01 |
| x_7 | .525000E+01 | .100686E+01 | .528600E+01 | .528600E+01 |

Table 8 gives the values of the constraints before and after scaling.

TABLE 8 PROBLEM 5 : Values of the Constraints Before and After Scaling

| Constraint No. | Initial Value | Limit | After Scaling |
|----------------|---------------|--------|---------------|
| 1 | .6882 | 1.0 | .6909 |
| 2 | .4503 | 1.0 | .4592 |
| 3 | .4061 | 1.0 | .3856 |
| 4 | .0714 | 1.0 | .0721 |
| 5 | 1049.6000 | 1100.0 | 1099.9000 |
| 6 | 867.6120 | 850.0 | 850.0560 |
| 7 | 16.8750 | 40.0 | 15.5068 |
| 8 | -4.1333 | -5.0 | -5.1429 |
| 9 | 4.1333 | 12.0 | 5.1429 |
| 10 | 3.1000 | 3.6 | 3.6000 |
| 11 | -3.1000 | -2.6 | -3.6000 |
| 12 | 0.7500 | 0.8 | .7000 |
| 13 | -0.7500 | -0.7 | -.7000 |
| 14 | 22.5000 | 28.0 | 22.1526 |
| 15 | -22.5000 | -17.0 | -22.1526 |
| 16 | 7.8000 | 8.3 | 7.3000 |
| 17 | -7.8000 | -7.3 | -7.3000 |
| 18 | 7.8000 | 8.3 | 7.6750 |
| 19 | -7.8000 | -7.3 | -7.6750 |
| 20 | 3.4000 | 3.9 | 3.3475 |
| 21 | -3.4000 | 2.9 | -3.3475 |
| 22 | 5.2500 | 5.5 | 5.2860 |
| 23 | -5.2500 | -5.0 | -5.2860 |
| 24 | 0.8974 | 1.0 | .9481 |
| 25 | 0.0840 | 1.0 | 1.0052 |

These five examples illustrate the power and mechanics of the scaling algorithm. Complete details of how to use this scaling algorithm are given in Reference 5.

6.0 SUMMARY AND CONCLUSIONS

This paper is organized into six sections, inclusive. The first two sections explain scaling and its intended purpose in the context of constrained mathematical optimization. They also contain a brief history and motivation for the development of the compound scaling algorithm.

The third section generalizes the simple scaling discussed earlier^(3,4) with additional observations and options. The subsection on classification of constraints contains a comprehensive description of most of the constraint types encountered in mathematical optimization and provides guidelines for effective implementation in scaling. The interaction formula for the scale factor was originally derived^(3,4) in the context of application to beam problems. The beam deformation can be separated into axial and bending parts when there is no coupling between them. The axial part is characterized by linear behavior while the bending part is governed by nonlinear behavior. This paper derives the basis for this concept in the context of general mathematical functions while extracting a much simpler relationship. This relationship in turn leads to the compound scaling algorithm discussed in Section 4. In essence the compound scaling algorithm is an elegant extension of simple scaling to multiple constraints.

The compound scaling algorithm requires basically two matrices. The first is the constraint values matrix, \bar{z} . The dimension of this matrix is $s \times 1$, where s represents the number of active constraints. The second is a matrix of constraint gradients (table). N . The dimension of this matrix is $n \times s$, where n is the number of variables and s represents the number of active constraints. These two matrices are generally obtained from an analysis of the system. From this information two additional matrices A and t are generated. The dimensions of these two matrices are identical to those of the N matrix. The first is designated as the matrix of possible scale factors (table), while the second

is called the scale factor assignment matrix (table). The information from the last three tables is the basis for an orderly derivation of individual scale factors for all the variables. Then the relationship between the scaled and the unscaled variables is given by

$$x_i = \Lambda_i x_i \quad i = 1, 2 \dots n$$

The scaled variables are closer to the constraint boundary.

Section five contains the results of application to simple mathematical functions. The first two examples show that the scaling can be exact no matter how far the design is from the constraint boundary. This means that the constraint boundary can be reached in a single step. In these two examples the parameter μ remains constant throughout the n -dimensional space, and the scaling is exact. The third example shows that this is not the case when the parameter μ varies in the n -dimensional space. In such cases scaling would be significantly faster than any search. The fourth example illustrates application of the compound scaling algorithm to a single constraint. The last example is an application to multiple constraints. These examples not only illustrate the advantages and the limitations of the scaling algorithm but also the details of application.

The characteristics that make this compound scaling algorithm most attractive are outlined as follows:

1. The constraint surface can be reached very rapidly from anywhere in the n -dimensional space.
2. All the decisions in scaling are tied to the values of two key parameters μ and β .
3. Determination of these two parameters is extremely simple and computationally insignificant when the constraint and gradient information are available from an analysis of the system.
4. Selection of appropriate initial values for the variables is no longer an issue.

5. The need for directional searches and move limits is eliminated.
6. The step size determination is automatic and is tied to the values of the parameters μ and β .
7. Algorithm implementation is simple and straightforward, because computational overhead beyond an analysis of the system is almost insignificant.

Future research in optimization should concentrate on the behavior of the mathematical functions through the study of these non-dimensional parameters.

This compound scaling algorithm constitutes a significant breakthrough in the evaluation of the optimality criteria approach for large scale optimization (with thousands of variables and constraints) in a multidisciplinary setting.

REFERENCES

1. Venkayya, V.B., Khot, N.S., and Reddy, V.S., (1969): "Energy Distribution in an Optimum Structural Design," AFFDL-TR-68-156.
2. Venkayya, V.B., (1971): "Design of Optimum Structures," J. Computers and Structures -1, pp 265-309.
3. Venkayya, V.B., (1988). "A Further Generalization of Optimality Criteria Approaches in Structural Optimization", Proceedings of the International Conference on Computational Engineering Science, April 10-14, 1986, Atlanta, Ga., pp 46i.1-15
4. Venkayya, V.B., (1989). "Optimality Criteria: A Basis for Multidisciplinary Design Optimization," J. Computational Mechanics -4
5. Venkayya, V.B., Tischler, V.A., and Kolonay, R.M., (1989), "Implementation of a Generalized Optimality Criteria Method," AFWAL-TR-89-
6. Azarm, S., Li, W.C., (1988), "A Multilevel Optimization Design Procedure Using Global Monotonicity Analysis," ASME Advances in Design Automation DE Vol 14, pp 115-124.